

Herramienta Informática para la Aplicación de Técnicas Metaheurísticas a la Programación de la Producción de Sistemas Job-Shop

Álvaro García Sánchez¹, Luis Miguel Arreche Bedia²

¹Ingeniero Industrial, U.D. Organización de la Producción. E.T.S.I. Industriales. Universidad Politécnica de Madrid. José Gutiérrez Abascal 2, 28006 Madrid. agsanchez@ingor.etsii.upm.es

²Doctor Ingeniero Industrial. U.D. Organización de la Producción. E.T.S.I. Industriales. Universidad Politécnica de Madrid. José Gutiérrez Abascal 2, 28006 Madrid. arreche@ingor.upm.es

RESUMEN

La programación de la producción es uno de los principales campos de aplicación de las técnicas metaheurísticas. Se ha desarrollado una herramienta informática para el estudio de la aplicación de tres técnicas metaheurísticas en problemas de programación de la producción. Las técnicas consideradas son: la búsqueda tabú, los algoritmos genéticos y el recocido simulado. En el presente trabajo se describen los rasgos más relevantes de esta herramienta informática, con la que será posible realizar estudios posteriores que permitan evaluar la eficacia y la eficiencia de las técnicas metaheurísticas consideradas para la resolución de distintos problemas de programación de la producción.

1. Introducción

Desde los años setenta han ido apareciendo diferentes métodos de resolución de problemas combinatorios, llamados técnicas metaheurísticas.

Estas técnicas metaheurísticas no garantizan la obtención de soluciones óptimas y, en general, tampoco permiten estimar la proximidad de dichas soluciones al óptimo, pero permiten estudiar problemas de gran complejidad de una manera sencilla y obtener soluciones suficientemente buenas en tiempos razonables.

Las técnicas metaheurísticas más extendidas son las siguientes: los algoritmos genéticos, propuestos por Holland en 1975; la búsqueda tabú, propuesta por Glover en 1986; el recocido simulado, propuesto por Kirkpatrick, Gelatt y Vecchi en 1983. Las redes neuronales, también incluidas en las técnicas metaheurísticas, son de naturaleza diferente a las tres anteriores y no han sido objeto de estudio denle el presente trabajo.

La lógica de las técnicas metaheurísticas es similar: el punto de partida es una solución (o conjunto de soluciones) que típicamente no es óptima. A partir de ella se obtienen otras soluciones, de entre las cuales se elige una que satisface a algún criterio, a partir de la cual comienza de nuevo el proceso. Este proceso se detiene cuando se cumple alguna condición establecida previamente.

Los campos de aplicación de los técnicas metaheurísticas son numerosos (electrónica, telecomunicaciones, electromagnetismo, etc.), entre ellos se encuentra la ingeniería de organización, con distintas aplicaciones, por ejemplo, a problemas de programación de la producción, diseño de rutas de distribución física, etc.

2. Descripción del problema

El problema de programación de la producción de un sistema de tipo job-shop considerado este trabajo consiste en obtener un programa que establezca cómo se procesan n órdenes en m máquinas en el menor tiempo posible. El problema queda definido de la siguiente manera:

- Existen n órdenes $O_1...O_n$ y m máquinas $M_1...M_m$.
- Cada orden O_i debe ser sometida a un total de m operaciones conforme a un determinado orden, la secuencia de estas operaciones se denomina ruta.
- Cada operación de la orden O_i sólo se puede realizar en una máquina; con un tiempo de proceso tp_{ij} .
- Para cada orden no existen dos operaciones que se realicen en la misma máquina.
- Una vez que una orden está siendo procesada no puede detenerse. Es decir, no se pueden procesar parcialmente las órdenes.
- Las máquinas no pueden procesar más de una orden simultáneamente.

El problema queda definido cuando se conocen:

- El número de órdenes, n
- El número de máquinas, m
- Las rutas: rij es la máquina que procesa a la orden i en la posición j -ésima, con $i=1...n$, $j=1...m$.
- Los tiempos de proceso: tp_{ij} tiempo que tarda la máquina j en procesar la orden i , con $i=1...n$, $j=1...m$.

Un programa debe determinar los instantes iniciales y finales de cada una de las operaciones que realiza cada una de las máquinas.

La descripción anterior del problema supone que los tiempos de proceso en las distintas máquinas son deterministas. En muchas ocasiones esta hipótesis puede no ser admisible y, por tanto, es necesario diseñar un modelo en el que se contemple esta posibilidad.

Por este motivo, la aplicación informática permite también estudiar problemas no deterministas. En este caso, se ha supuesto que los tiempos de proceso siguen una distribución beta, con parámetros α y β , distribuida entre los valores a y b .

Los parámetros que se seleccionen deben ser tales que la función de distribución tenga una forma semejante a la de la figura 1: los tiempos de proceso de cada operación tienen un tiempo técnico mínimo por debajo del cual, para una tecnología y una organización determinadas, no es posible realizar dicha operación. Además, a pesar de que es posible obtener tiempos elevados con respecto a la media, la probabilidad de estos es muy pequeña. Por último, la probabilidad de que los tiempos de proceso sean muy elevados es igual a cero. Por tanto, la función de distribución beta se considera adecuada para representar los tiempos de proceso en contexto no determinista.

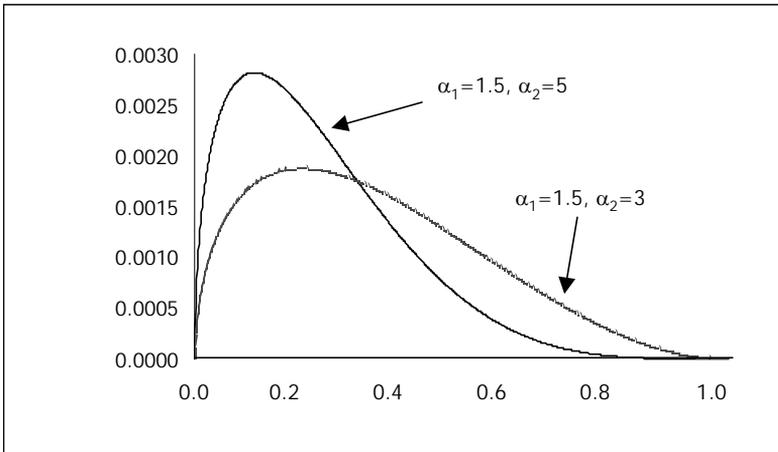


Figura 1: Tiempos de proceso en contexto no determinista

Cuando se han estudiado problemas en contexto no determinista, se ha supuesto que se conocen los valores medios de dichos tiempos ($\bar{t}p_{ij}$) y los parámetros α_1 y α_2 que determinan la forma que adopta la función de distribución. Además se asume la hipótesis de que el tiempo técnico mínimo es igual al 90% de la media. Es decir, $a_{ij} = 0.9 \cdot \bar{t}p_{ij}$. Con esto, el valor del máximo tiempo de proceso es:
$$b_{ij} = a_{ij} + \left(\bar{t}p_{ij} - a_{ij} \right) \cdot \frac{\alpha_1 + \alpha_2}{\alpha_1}$$

Cuando la aplicación trabaja en contexto determinista, la comparación de las diferentes soluciones es sencilla y consiste simplemente en comparar los respectivos valores de la función objetivo.

En contexto no determinista, para la comparación de dos soluciones se efectúa un número de repeticiones definido en cada ocasión (n_r) y se realiza un contraste de hipótesis para identificar si las medias de las variables aleatorias “tiempo total de proceso” de las dos soluciones son significativamente distintas o no. Todas las comparaciones se realizan con un nivel de confianza de $\alpha = 0.05$.

Cuando lo que se pretende es seleccionar la mejor o las k mejores de entre un conjunto de soluciones, se ha elegido un método sencillo que no garantiza plenamente la selección de la mejor de las soluciones de cada conjunto considerado. La justificación a esta decisión se basa en que, de hecho, la filosofía de las técnicas metaheurísticas consiste en obtener soluciones aceptables en tiempos razonables y, por tanto, no tiene sentido invertir recursos en perfeccionar el método de selección de la mejor alternativa, siempre y cuando la que se escoja sea una buena alternativa, similar a la mejor (si no la mejor). Cuando las diferencias entre las soluciones son muy significativas el método empleado es válido, y cuando las diferencias no lo son, en caso de no seleccionar la mejor se seleccionará una muy próxima a la mejor. Por tanto, el método se considera adecuado en términos de la bondad de la solución seleccionada.

3. Técnicas metaheurísticas empleadas. Aplicación informática

A continuación se describen brevemente algunas características de la aplicación informática. Y posteriormente se ponen de relieve los aspectos más significativos de la forma que adoptan las técnicas elegidas para el problema propuesto.

3.1. Herramienta informática

Para la resolución del problema de la producción planteado anteriormente, se ha diseñado una aplicación informática en Visual Basic 6.0. En general, la aplicación será válida para la resolución de cualquier problema que admita una representación como la descrita anteriormente. Quizá, para un problema distinto sólo sea necesario definir la función objetivo que será distinta a la del problema considerado en este trabajo.

Dicha aplicación consta de cuatro módulos. Un módulo común para todas técnicas metaheurísticas en el que se definen los datos del problema que se desea estudiar y otros tres módulos, uno por cada una de las técnicas metaheurísticas, en el que se definen las particularidades de cada una de ellas.

3.2. Aspectos comunes a las tres técnicas

La representación de la solución es una matriz de tantas filas como órdenes (n) y tantas columnas como máquinas (m). Los elementos de cada una de las columnas determinan la prioridad con la que se procesan los órdenes en cada máquina cuando dichas órdenes solicitan simultáneamente una determinada máquina.

De forma más precisa, cada elemento X_{ij} de la matrices un número entero comprendido entre 1 y n y representa una de las n órdenes. En la máquina j , el trabajo correspondiente al elemento X_{ij} tiene prioridad frente a los trabajos correspondientes a los elementos X_{kj} con $k > i$.

Esta representación de la solución es tal que cualquier matriz que contenga los n primeros números naturales en cada una de las columnas (independientemente del orden de los mismos) es una solución factible del problema. Sin embargo, la relación no es biunívoca, ya que en ocasiones algunas de las prioridades de algunas columnas no operarán y, por tanto, existirán matrices que conducirán a la misma programación de la producción y ofrecerán el mismo valor de la función objetivo. Por otro lado, la representación sí contiene todo el espacio de soluciones (aunque con redundancias como se ha señalado). Conviene notar que el espacio de soluciones no contiene a aquellas posibles soluciones en las que alguna máquina, en algún momento permanece ocioso a la vez que una orden requiere ser procesada precisamente por esa máquina

Cualquiera de las técnicas metaheurísticas detiene el proceso de búsqueda cuando se cumple algún criterio. La aplicación informática permite seleccionar uno de los siguientes criterios en cualquiera de las tres técnicas:

- Número máximo de iteraciones.
- Tiempo máximo de ejecución.
- Número de iteraciones sin mejorar la solución.
- Valor de la función objetivo.

La aplicación permite generar de forma aleatoria soluciones de partida, asignando de manera aleatoria los n primeros números naturales a las cada una de las columnas. Alternativamente, permite introducir una solución de partida obtenida a través de otra técnica, de un heurístico, etc.

3.3. Aspectos relativos a la búsqueda tabú

Dada una solución, el vecindario se puede construir de dos maneras diferentes: mediante movimientos de intercambio y mediante movimientos de inserción. Se debe elegir uno de los dos tipos de movimientos para generar el vecindario.

- Movimientos de intercambio. Con este tipo de movimientos, las soluciones vecinas son aquellas que se obtienen intercambiando la posición que ocupan dos elementos cualesquiera pertenecientes a una misma columna.

Se introducen dos posibilidades para restringir el tamaño del vecindario y reducir el tiempo de computación: en primer lugar, permitir sólo aquellos intercambios que se realicen entre elementos de la matriz cuya distancia no supere un determinado valor y, en segundo, realizar los intercambios correspondientes a un determinado número de columnas y no a todas. Ambas restricciones se pueden adoptar simultáneamente.

- Movimientos de inserción. Un movimiento de inserción consiste en el desplazamiento de un elemento de la matriz a una fila distinta de la que ocupa dentro de la misma columna. Algunos de los elementos de la misma columna deben ser desplazados manteniendo sus posiciones relativas.

Igualmente, es posible restringir el vecindario generado mediante movimientos de inserción bien acotando la distancia a la que se puede desplazar un elemento, o bien realizando los movimientos de inserción sólo en algunas columnas.

La búsqueda incorpora una memoria a corto plazo que consiste en una lista tabú de una longitud determinada, L . La lista tabú contiene la información que permite evaluar en cada iteración las soluciones que son tabú y las que no lo son. La aplicación permite establecer que la búsqueda trabaje con movimientos tabú o que lo haga con atributos tabú: movimientos tabú (son aquellos movimientos realizados en las últimas L iteraciones) y atributos tabú (son las características de la solución que se modifican en cada iteración. Dependen del tipo de movimiento seleccionado).

La forma que adopta la lista tabú es siempre una matriz, con L o $2L$ columnas y con diferente número de filas según del caso del que se trate.

El criterio de aspiración seleccionado viene determinado por la función objetivo y es de carácter global, esto es, una solución dejará de ser tabú si su valor de la función objetivo es mejor que el de la mejor solución encontrada hasta el momento.

3.4. Aspectos relativos al algoritmo genético

En primer lugar se debe definir el número de individuos N_{ind} que constituyen cada población. No se ha incluido ninguna restricción con respecto al valor que puede adoptar este parámetro.

La función de adaptación de un individuo se puede definir de dos formas distintas: el opuesto de la función objetivo: $F_{adap} = -F_{obj}$ y el inverso de la misma: $F_{adap} = 1/F_{obj}$

Se han considerado los siguientes tipos la selección: selección proporcional, selección determinista, selección normalizada, selección por clasificación y selección por torneo.

El número de individuos seleccionados en cada iteración es siempre un número par para poder realizar cruces por parejas para generar los individuos de la descendencia. El número de individuos seleccionados se determina como proporción sobre el total de individuos de la población de partida.

Asimismo, se han tenido en cuenta cuatro tipos de cruce diferentes: intercambio de columnas, intercambio de filas, intercambio de órdenes y cruce de columnas homólogas (se seleccionan una o varias columnas, según se determine, de forma aleatoria. Para cada columna, se seleccionan también aleatoriamente dos puntos de corte, de manera que los elementos de los descendientes comprendidos entre dichas posiciones se intercambian). La selección de las columnas, filas o trabajos tiene lugar de manera aleatoria. En ocasiones es necesario realizar alguna modificación para garantizar la consistencia de la solución.

La mutación opera sobre cada uno de los elementos con una probabilidad p_m . La mutación consiste en asignar un nuevo valor al elemento de la matriz y modificar la misma para evitar inconsistencias.

Los operadores de reducción que se han considerado para obtener una nueva población a partir de los individuos de la población inicial y de sus descendientes son los siguientes: reemplazo de todos los individuos por sus descendientes, selección de los mejores individuos y selección de los N_0 mejores de individuos de entre los descendientes y los N_{ind} mejores de entre los padres.

3.6. Aspectos relativos al recocido simulado

La aplicación permite seleccionar uno de los siguientes patrones: función aritmética (la temperatura se comporta como una progresión aritmética de diferencia d), función geométrica (la temperatura se comporta como una progresión aritmética de razón r) y función aritmético-geométrica ($T_i = T_{i-1} - (i-1) \cdot r$).

La forma de generar una solución vecina a partir de una dada puede realizarse de dos maneras distintas: cambiando dos elementos de forma aleatoria o intercambiando dos columnas elegidas de forma aleatoria.

La aplicación incorpora la opción de explorar una sola solución en cada temperatura o bien explorar L soluciones vecinas en cada temperatura. En este caso se debe definir el mínimo número de soluciones peores aceptadas para que una iteración en la que no se ha encontrado una solución mejor no compute como una iteración en la que no se ha mejorado la solución.

El criterio de detención consiste en finalizar la búsqueda cuando no se ha mejorado la solución durante un número determinado de iteraciones. Este criterio de detención es similar al del recocido simulado en el sentido de que no todas las iteraciones en las que no se ha mejorado la solución se consideran como tales en su.

4. Conclusiones

Las técnicas metaheurísticas consideradas aplicadas a los problemas de programación de la producción cubren un conjunto de posibilidades amplio. Con la aplicación informática desarrollada se han realizado estudios en problemas concretos tanto en contexto determinista como no determinista y se ha comprobado que puede resultar una herramienta de gran utilidad para estudiar el comportamiento de distintas configuraciones de las técnicas metaheurísticas para la resolución de diferentes tipos de problemas. Estas investigaciones futuras permitirán extraer conclusiones de carácter general. .

5. Bibliografía

- Modern Heuristic Techniques for Combinational Problems. REEVES, COLIN R., McGraw-Hill, 1995. Guildford.
- Iterative Computer Algorithms with Applications in Engineering. Solving Combinatorial Optimization Problems. SAIT M. SADIQ y YOUSSEF HABIB. IEEE Computer Society, 1999.
- Intelligent Optimisation Techniques. PHAM D.T. y KARABOGA D., Springer, 2000. London.
- Genetic Algorithms and Grouping Problems. Emanuel Falkenauer. John Wiley and Sons, 1998. Cheichester.
- Godberg, David E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA.
- Holland, John H. (1975). Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor,